LLM-guided Plan and Retrieval: A Strategic Alignment for Interpretable User Satisfaction Estimation in Dialogue

Sangyeop Kim[1,2], Sohhyung Park[1], Jaewon Jung[1], Jinseok Kim[1], Sungzoon Cho[†1]
[1] Seoul National University
[2] Coxwave
{sy917kim, sohhyung, wjdwodnjs302, jsk0821}@bdai.snu.ac.kr
zoon@snu.ac.kr

Abstract

Understanding user satisfaction with conversational systems, known as User Satisfaction Estimation (USE), is essential for assessing dialogue quality and enhancing user experiences. However, existing methods for USE face challenges due to limited understanding of underlying reasons for user dissatisfaction and the high costs of annotating user intentions. To address these challenges, we propose PRAISE (Plan and Retrieval Alignment for Interpretable Satisfaction Estimation), an interpretable framework for effective user satisfaction prediction. PRAISE operates through three key modules. The Strategy Planner develops strategies, which are natural language criteria for classifying user satisfaction. The Feature Retriever then incorporates knowledge on user satisfaction from Large Language Models (LLMs) and retrieves relevance features from utterances. Finally, the Score Analyzer evaluates strategy predictions and classifies user satisfaction. Experimental results demonstrate that PRAISE achieves state-of-the-art performance on three benchmarks for the USE task. Beyond its superior performance, PRAISE offers additional benefits. It enhances interpretability by providing instance-level explanations through effective alignment of utterances with strategies. Moreover, PRAISE operates more efficiently than existing approaches by eliminating the need for LLMs during the inference phase.

1 Introduction

Dialogue systems play an increasingly crucial role in enabling users to interact with intelligent agents to fulfill their needs. **User Satisfaction Estimation** (USE), the process of predicting how satisfied a user is in dialogue interactions (Choi et al., 2019), is crucial for evaluating the quality of dialogue systems and ensuring a positive user experience (Bodigutla et al., 2019; Cai and Chen, 2020; Deng et al., 2022; Liang et al., 2021; Pan et al., 2022; Siro et al., 2023; Ye et al., 2023). Effective USE methods should not only accurately classify user satisfaction but also provide interpretable results to guide the improvement of dialogue systems. By understanding and quantifying user satisfaction, dialogue systems can be continuously improved to better meet user expectations.

The development of USE methods has evolved through three main approaches: content-based, dialogue act-based, and language model-based. Content-based methods, such as sentiment analysis (Song et al., 2019) and response quality assessment (Schmitt and Ultes, 2015), evaluate dialogue content to estimate user satisfaction. However, these methods often struggle to accurately capture user intentions and whether user goals are fulfilled.

Dialogue act-based methods incorporate dialogue acts, which represent user intentions at each turn (Chen et al., 2018; Stolcke et al., 2000; Ye et al., 2023; Yu et al., 2019), leveraging the relationship between these acts and user goal achievement (Deng et al., 2022). However, these methods often require complex pre-training procedures and accurate dialogue act labels, which can be challenging and time-consuming to obtain for real-world conversations.

Language model-based methods have shown a promising direction for USE. Lin et al. (2024) employs Large Language Models (LLMs) and iterative prompting to summarize dialogues and extract rubrics of user satisfaction from natural language utterances. However, this approach does not provide utterance-level interpretability and relies heavily on advanced LLMs like GPT-4 for the entire process, which can be expensive and impractical for large-scale applications.

In this paper, we introduce **PRAISE** (**P**lan and **R**etrieval **A**lignment for **I**nterpretable **S**atisfaction **E**stimation), a framework that leverages LLMs to generate and refine strategies for classifying user

[†]Corresponding author.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 10416–10430

satisfaction levels in conversational systems. These strategies serve as interpretable natural language criteria that indicate situations for identifying user satisfaction (SAT), dissatisfaction (DSAT) or neutrality (NEU) in dialogues. For instance, as shown in Figure 1, a strategy like "User thanks for solving problem quickly" can be an indicator of user satisfaction.

- **User thanks for solving problem quickly.**
- **User requests additional information.**
- **User accepts with "Yes"**

Figure 1: Examples of strategies for satisfaction

The framework consists of three key modules: **Strategy Planner**, **Feature Retriever**, and **Score Analyzer** (Figure 2). The Strategy Planner generates interpretable natural language strategies for classifying user satisfaction by leveraging effective and ineffective strategies stored in memory. The Feature Retriever quantifies the relevance between user utterances and generated strategies, converting the LLM knowledge into measurable features for analysis. The Score Analyzer evaluates the effectiveness of strategies and categorizes them according to their contributions to USE improvement. Through iterative refinement of this process, PRAISE aims to identify optimal strategies to maximize overall performance.

PRAISE achieves state-of-the-art performance on USE benchmark datasets like MWOZ, SGD, and ReDial, while providing interpretable results. Beyond its high performance and interpretability, PRAISE brings additional advantages, including efficient inference and scalable deployment. The key contributions of our paper are:

- We propose PRAISE, a novel LLM-powered framework that automates the generation and validation of strategies specifically designed for User Satisfaction Estimation.
- We present an approach that offers utterance-level interpretability, providing crucial insights for improving dialogue systems.
- We transform LLM knowledge into measurable features, enabling efficient and scalable inference without direct LLM usage.
- We demonstrate the effectiveness of PRAISE through extensive experiments on benchmark datasets, showing its robustness.

2 Problem statement

Consider a dataset D of conversations, where each conversation C_i is represented as a sequence of T utterance pairs between the user and the assistant, denoted as $C_i = [(U_{i,1}, A_{i,1}), ..., (U_{i,T}, A_{i,T})]$. Here, $U_{i,n}$ represents the n-th user utterance in the i-th conversation, and $A_{i,n}$ represents the corresponding assistant response. Each user utterance $U_{i,n}$ is associated with a satisfaction label $y_{i,n}$, which can take one of three values: SAT, DSAT, or NEU. The user satisfaction estimation problem can be formulated as learning a satisfaction estimator $\mathcal{E}$ that maps the current user utterance $U_{i,n}$ and the preceding conversation context to the corresponding satisfaction label $y_{i,n}$.

Satisfaction estimator ($\mathcal{E}$):

$$\{(U_{i,1}, A_{i,1}), ..., (U_{i,n-1}, A_{i,n-1}), U_{i,n}\} \rightarrow y_{i,n}$$

3 Method

3.1 PRAISE: Plan and Retrieval Alignment for Interpretable Satisfaction Estimation

The PRAISE framework is designed to address the challenges of interpretability and scalability in USE models. PRAISE consists of three key modules: *Strategy Planner*, *Feature Retriever*, and *Score Analyzer*. The Strategy Planner uses LLMs to formulate hypothetical strategies for classifying user satisfaction. These strategies are interpretable natural language criteria that indicate user satisfaction levels in dialogues. The Feature Retriever generates passages based on these strategies and compares them with user utterances to extract quantified relevance features between the utterances and strategies. The Score Analyzer uses these features to compute a user satisfaction score, which is then used to classify user satisfaction levels. As a result, strategies that enhance classification performance are incorporated into effective strategies (S_+), while others go into ineffective strategies (S_-).

3.1.1 Strategy Planner

The Strategy Planner generates strategies for classifying user satisfaction using LLMs. It requires a problem-defining prompt and two types of strategies from previous USE evaluations: effective strategies (S_+) and ineffective strategies (S_-). These S_+ and S_- guide the generation of new strategies to optimize overall performance, enhancing reasoning ability and avoiding redundancy. The

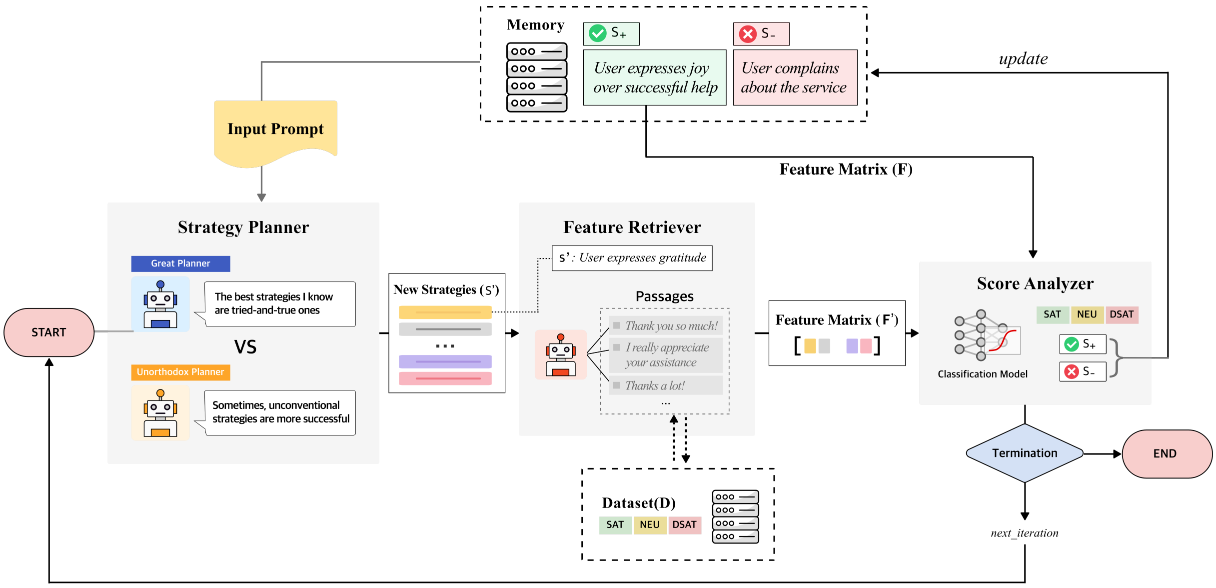


Figure 2: The overall framework of PRAISE.

planner then produces n_s strategies that define the scenarios for SAT, DSAT, and NEU. Initially, 3 to 5 human-defined strategies are provided as S_+. Through iterations, S_+ and S_- are updated based on improvements in user satisfaction classification.

We employ two distinct planner types: the **Great Planner** and the **Unorthodox Planner**. The great planner, operating at a lower temperature, generates strategies directly relevant to user satisfaction analysis. However, the great planner tends to generate consistent strategies when there are only minor changes in S_+ and S_-. To address this, we introduce the unorthodox planner which not only operates at a higher temperature but also receives specific prompts encouraging the generation of unconventional yet plausible strategies. The framework selects between these planners based on an exploration ratio (ϵ), choosing the unorthodox planner with this probability. If the validation score does not improve, the exploration ratio doubles to expand the search space. When improvement occurs, the ratio resets to its initial value ϵ.

3.1.2 Feature Retriever

The feature retriever quantifies the relationship between user utterances and strategies previously generated by the planner, aiming to identify the user utterances that best align with each strategy. To achieve this, it calculates the similarity between each strategy and individual user utterances. For effective similarity calculation, we adopt the retrieval method proposed by Gao et al. (2022a). The entire retrieval process consists of two stages: passage generation and feature retrieval.

Passage Generation In the first stage, the feature retriever generates k hypothetical user passage examples $p_{s'} = \{p_{s',1}, p_{s',2}, ..., p_{s',k}\}$ that correspond to each strategy s' produced by the planner. As illustrated in the feature retriever example in Figure 2, for a strategy like "User expresses gratitude", the passage generation step could produce examples such as "Thank you so much!", or "I really appreciate your help". This process creates plausible user passages for each strategy s'. Subsequently, these passages are transformed into embeddings $E_{p_{s'}} \in \mathbb{R}^{k \times d}$ by an embedding model, where d represents the embedding dimension. The generation of multiple passages for each strategy ensures capturing a diverse range of potential user expressions, enabling more accurate similarity calculations between strategies and actual user utterances.

Feature Retrieval To retrieve relevant features for each strategy, we compute the matrix product between the generated passage embeddings $E_{p_{s'}}$ and the embeddings of actual user utterances from the dataset, denoted as $E_u \in \mathbb{R}^{m \times d}$, where m is the number of utterances in the dataset. This step is crucial as it allows us to measure the similarity between our hypothetical strategy-based passages and the real user responses. The matrix product operation generates a relevance matrix $\mathcal{R} \in \mathbb{R}^{k \times m}$ for all combinations of generated passages and actual utterances. Each element in this matrix represents the similarity score between a generated passage and a real utterance. We then sum these relevance scores across all passages to obtain a single feature score for each strategy s':

$$f'_{s'} = \sum_{i=1}^{k} \mathcal{R}_i$$

The term $f'_{s'} \in \mathbb{R}^m$ represents the overall relevance score between a generated strategy (s') and each utterance. We then stack these scores for all newly generated strategies to form a feature matrix $F' \in \mathbb{R}^{m \times n_s}$, where n_s is the number of newly generated strategies. By quantifying strategies as similarity scores for each utterance, we create structured data that conventional machine learning models can effectively process and analyze.

3.1.3 Score Analyzer

The score analyzer evaluates the effectiveness of generated strategies and trains the model to classify user satisfaction level. This module refines the strategy set and improves the overall performance of the user satisfaction.

We compute a baseline score (score_0) using the feature matrix F from the previous best strategies (S_+). We employ logistic regression as the final classification model ($\mathcal{M}$) and evaluate its performance on the validation set. We add new features (F') to the existing ones (F) column by column and evaluate using $\mathcal{M}$ to compare the resulting score with score_0. Strategies that make the score better go into S'_+, others into S'_-. We then update our overall strategy sets S_+ and S_-.

However, continuously increasing the number of strategies can lead to longer prompts and reduced interpretability. To solve this issue, we implement top-k strategy selection, which identifies and uses only the most useful strategies as effective (S_+) to optimize the combination. In the logistic regression model, the absolute values of the coefficients for all labels are summed, and the top-k features with the largest values are selected. When using other models, model-specific importance measures (Breiman, 2001) or model-agnostic importance calculation (Lundberg and Lee, 2017; Ribeiro et al., 2016) can be employed. The pseudo-code of the score analyzer is shown in **algorithm 1**.

3.2 Inference

During the inference phase, PRAISE employs its trained model and refined strategies to classify user satisfaction for individual utterances in new dialogues. The feature retriever operates similarly to the training phase but uses only the final set of effective strategies (S_+). Specifically, the feature vector F is calculated using the passage embeddings E_p from the training phase and the new utterance embeddings E_u^{test}. This approach of reusing the passages has two key advantages: it ensures consistency with the optimal results and reduces computational costs by eliminating the need for additional passage generation. This feature vector is then passed to the score analyzer, where the logistic regression model predicts the user satisfaction level for the utterance. This approach enables PRAISE to evaluate user satisfaction efficiently without LLM inference, ensuring scalability through simple models. Additionally, the feature vector F and S_+ provide utterance-level interpretability.

Algorithm 1 Selective Feature Addition Based on Score Improvement with Top-k Selection

```
S'_+, S'_- ← ∅                                   ▷ Initialize strategies set
length ← ColumnCount(F')                          ▷ Get columns in F'
score_0 ← Evaluate(M, F)                          ▷ Evaluate original features
for j = 0 to length do
    score ← Evaluate(M, F ⊕ F'[:, j])
    improvement ← score − score_0
    if improvement > 0 then
        S'_+ ← S'_+ ∪ {S'[j]}
    else
        S'_- ← S'_- ∪ {S'[j]}
    end if
end for
coef ← GetCoefficients(M)                         ▷ Get coefficients for all classes
Let L be the number of classes in the classification problem
importance_i = Σ_{l=1}^{L} |coef_{i,l}|,   ∀i ∈ {1, ..., |S'_+|}
Sort S'_+ in descending order based on importance
S'_+ ← first k elements of sorted S'_+
return S'_+, S'_-
```

4 Experiments

4.1 Experimental setup

Dataset	MWOZ	SGD	ReDial
# Conversations	1,000	1,000	1,000
# Utterances	21,706	24,148	16,616
# Labeled utterances	8,439	9,316	7,304
Label distribution (SAT / NEU / DSAT, %)	40.4 / 32.2 / 27.4	47.6 / 30.2 / 22.2	49.5 / 26.9 / 23.6

Table 1: Statistics of datasets.

Datasets and metrics We use three task-oriented dialogue datasets (Li et al., 2016; Wu et al., 2019): ReDial (Li et al., 2018) for movie recommendations, and SGD (Rastogi et al., 2020) and MWOZ (Budzianowski et al., 2018) for general scenarios such as bookings and information requests. User satisfaction is annotated as 'SAT' when the system effectively resolves user queries and achieves their goals, 'DSAT' when it fails to meet user needs or provides irrelevant responses, and 'NEU' when

Model	MWOZ				SGD				ReDial			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
HiGRU (Jiao et al., 2019)	44.6	43.7	44.3	43.7	50.0	47.3	48.4	47.5	46.1	44.4	44.0	43.5
BERT (Devlin et al., 2018)	46.1	45.5	47.4	45.9	56.2	55.0	53.7	53.7	53.6	50.5	51.3	50.0
USDA (Deng et al., 2022)	49.9	49.2	49.0	48.9	61.4	60.1	55.7	57.0	57.3	54.3	52.9	53.4
ASAP (Ye et al., 2023)	56.6	55.1	54.9	54.9	64.4	62.7	62.5	62.5	62.9	60.2	60.4	60.0
GPT-3.5-turbo	36.3	57.3	43.9	30.9	51.6	40.9	43.3	37.1	55.1	47.9	44.3	39.7
(+ 3-shots)	42.2	42.2	43.0	41.6	50.3	48.4	48.0	47.7	54.4	53.8	47.1	46.5
GPT-4 (+ 3-shots)	45.8	46.0	45.4	43.8	52.8	48.7	43.3	42.7	53.5	51.0	45.4	45.1
SPUR (Lin et al., 2024)	49.8	**67.1**	40.6	36.4	48.4	40.8	36.2	32.1	56.1	56.3	47.3	42.3
PRAISE (30)	**60.3**	55.8	**59.9**	56.7	64.5	61.6	63.0	62.2	**66.0**	64.4	63.6	63.8
PRAISE (50)	**60.3**	56.6	59.8	**57.3**	**65.6**	**63.0**	**63.6**	**63.2**	**66.0**	**64.5**	**63.7**	**63.9**

Table 2: Performance of models on MWOZ, SGD, and ReDial datasets. Numbers in parentheses after PRAISE indicate the maximum number of features used. **Bold** is the best performance, while underlined is the second-best.

it partially fulfills the request (Sun et al., 2021). We have converted the original five-level ratings (1-5) to three satisfaction classes: DSAT (average rating < 3), NEU (average rating = 3), and SAT (average rating > 3), which is more practical for real-world applications and aligns with previous studies (Deng et al., 2022; Ye et al., 2023). The dataset is split into train, validation, and test sets with an 8:1:1 ratio, excluding dialogues fewer than two turns. Table 1 presents the dataset statistics. We use the same evaluation metrics as in previous works (Deng et al., 2022; Ye et al., 2023), including Accuracy (Acc), macro-averaged Precision (P), Recall (R), and F1-score (F1).

Baselines We compare our proposed method with the following baseline models:

- **HiGRU** (Jiao et al., 2019) utilizes two Bidirectional GRUs (Chung et al., 2014) structures to capture user utterances and context information.
- **BERT** (Devlin et al., 2018) estimates user satisfaction by taking the concatenation of previous dialogue context and the user's last utterance, separated by a [SEP] token, as input.
- **USDA** (Deng et al., 2022) pre-trains on dialogue patterns leading to satisfaction using pseudo-labels, then employs an Attentive GRU model to estimate user satisfaction.
- **ASAP** (Ye et al., 2023) combines BERT with a Hawkes process (Mei and Eisner, 2017; Xiao et al., 2017) to better capture the temporal dynamics of the conversation.
- **GPT family** includes GPT-3.5[1] and GPT-4 (Achiam et al., 2023) models, which evaluate satisfaction based on instructional prompts in various settings such as zero-shot and few-shot learning.
- **SPUR** (Lin et al., 2024) uses iterative prompting with LLMs to generate rubrics from conversations, which are then used to evaluate user satisfaction.

The baseline SPUR model only predicts SAT/DSAT, excluding NEU. Therefore, we define a range for the satisfaction score ($-k \sim k$) and predict the user satisfaction as NEU if it falls within that range. We set the k-value that maximizes the F1-score for each validation set.

Implementation details PRAISE implements a strategy planner using GPT-4 (gpt-4-1106-preview) for generating strategies and a feature retriever employing GPT-3.5-turbo (gpt-3.5-turbo-0125) for passage generation. Both modules handle five strategies (n_s) and passages (k) respectively. The exploration ratio (ϵ) is set to 0.1. For text embedding, we use OpenAI's text-embedding-3-large[2] model with 1024 dimensions. The training process involves 50 iterations using macro-F1, with early stopping if the validation score fails to improve for five consecutive iterations. For the logistic regression component, we employed l2 penalty with C=100. To ensure convergence, max_iter was set to 500 for MWOZ and 700 for ReDial and SGD. All main experiments were conducted using NVIDIA H100 (80GB), and for inference speed experiments in Section 4.4, we additionally used GTX 1080Ti (11GB).

4.2 Main Results

Table 2 demonstrates that PRAISE achieves state-of-the-art performance across most metrics and datasets in user satisfaction estimation. PRAISE

[1]https://platform.openai.com/docs/models#gpt-3-5-turbo

[2]https://platform.openai.com/docs/guides/embeddings

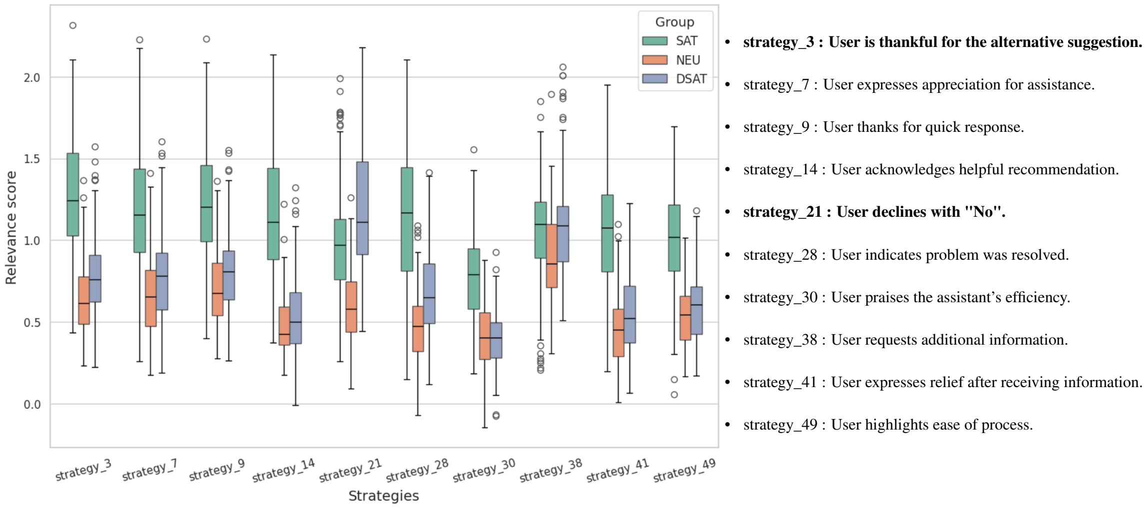


Figure 3: Box-plot of relevance scores for strategies in the SGD dataset.

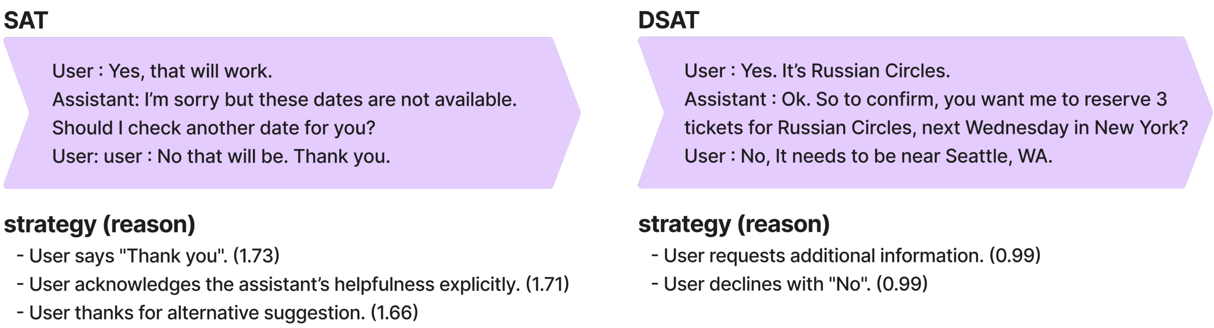


Figure 4: Strategies as interpretable reasons for predicting satisfaction.

obtains the highest F1-scores of 57.3%, 63.2%, and 63.9% on the MWOZ, SGD, and ReDial datasets, respectively. Additionally, it achieves the best accuracy scores of 60.3%, 65.6%, and 66.0% on MWOZ, SGD, and ReDial, respectively. Among the baselines utilizing Large Language Models (LLMs), including GPT-3.5-turbo, GPT-4, and SPUR, performance is relatively poor even with 3-shot learning. This suggests that direct application of LLMs, without task-specific fine-tuning or adaptation, may not be sufficient for accurate user satisfaction estimation. Notably, SPUR shows suboptimal performance despite being the most recent approach. This indicates that evaluating user satisfaction across entire conversation sessions may not be well-suited for utterance-level satisfaction assessment. Among the baseline models, ASAP demonstrates the best performance, though still not surpassing PRAISE.

4.3 Interpretability

The feature matrix F contains relevance scores that show how each utterance relates to the generated strategies. These scores provide interpretability by indicating which strategy is most relevant to a given utterance.

Figure 3 shows the relevance score distribution across satisfaction levels, illustrating that certain strategies effectively distinguish between SAT and DSAT. For example, "User is thankful for the alternative suggestion." (strategy_3) and "User declines with 'No'." (strategy_21) exhibit higher values for SAT and DSAT, respectively. These strategies provide clear explanations on distinguishing satisfaction levels in the current dataset.

The relevance scores for individual utterances provide insights into the reasons behind the predicted satisfaction level. Figure 4 demonstrates how these scores of the last utterance in a conversation can explain why it was classified as SAT, NEU, or DSAT. Additional examples are provided in Appendix B.

4.4 Scalability

In this study, we evaluate the scalability of the PRAISE model by comparing its inference time

Embedding Model	MWOZ				SGD				ReDial			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
mxbai-large (Sean Lee, 2024)	56.6	56.9	52.4	53.2	61.0	59.2	60.9	59.6	64.3	62.1	63.0	62.4
bge-large-en (Xiao et al., 2023)	57.2	56.7	52.7	53.3	61.7	59.9	61.6	60.3	64.7	**62.5**	**63.5**	**62.8**
gte-large-en (Li et al., 2023)	57.9	57.7	54.2	54.9	61.9	59.8	58.6	59.0	64.3	62.3	63.2	62.4
text-embedding-3-large	**60.3**	**56.6**	**59.8**	**57.3**	**65.6**	**63.0**	**63.6**	**63.2**	**66.0**	64.5	63.7	63.9

Table 3: Satisfaction classification performance with different embedding models

with the current state-of-the-art model, ASAP. A shorter inference time indicates better scalability, as it implies efficient utilization of computational resources. To evaluate inference times across different GPU environments, we test these models on high-end (NVIDIA H100) and low-end (NVIDIA GTX 1080Ti) setups, repeating each experiment 30 times for reliability. GPT-based models are excluded due to their lower performance and significantly longer inference times. As shown in Figure 5, the ASAP model is slower than the others on the low-end GPU, while the API-based model (i.e., text-embedding-3-large) is the fastest. In the high-end GPU environment, ASAP runs faster than the API-based model but remains slower than the other models. This shows that PRAISE with API-based model are the most effective in lower-spec GPU environments. Even without API access, PRAISE with other models are more efficient than ASAP. Also, PRAISE demonstrates high scalability by eliminating the need for LLMs during inference, utilizing only a pre-trained logistic regression model, the passage embedding matrix E_p, and an embedding model.

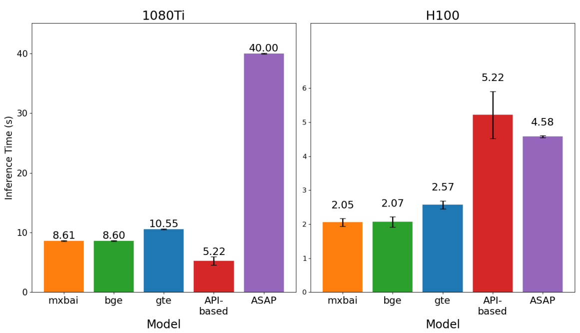


Figure 5: Comparison of inference time between PRAISE with various embedding models and ASAP

4.5 Ablation study

We conducted an ablation study on the techniques used in each module. We performed 20 repetitions of the experiment for each setting, starting from the initial strategies.

Unorthodox planner The probability of using the unorthodox planner is determined by the exploration ratio (ϵ). This ratio improves the final results by generating strategies that the great planner cannot conceive. Figure 6 shows that a 0.1 exploration ratio achieved higher maximum F1 scores in ReDial and MWOZ datasets. The SGD dataset did not show significant differences across various exploration ratios.

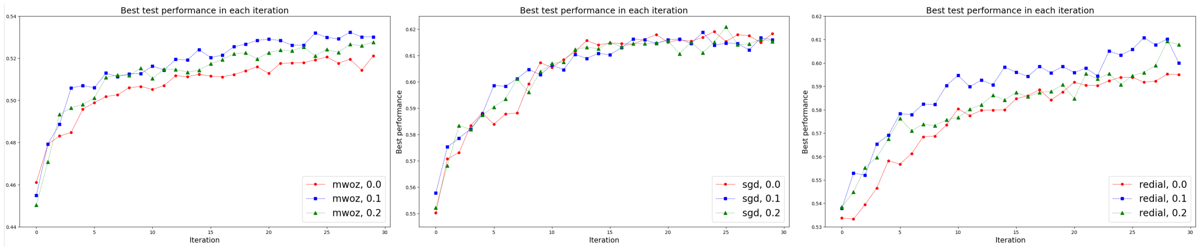
Embedding model We tested PRAISE with various embedding models to verify its robustness and adaptability. Our results are consistent across different embeddings (Table 3). The text-embedding-3-large model we employ in PRAISE exhibited the best performance among the embeddings tested. Notably, other embedding models we tested also delivered competitive results compared to text-embedding-3-large. These findings suggest that the inference stage could potentially eliminate the reliance on API-based embedding models, leading to significantly more cost-effective and time-efficient model operation.

Passage generation Our study utilizes a retrieval process based on passage generation inspired by (Gao et al., 2022a). This approach performed significantly better than using embeddings alone (Table 4).

Model	MWOZ		SGD		ReDial	
	Acc	F1	Acc	F1	Acc	F1
Not Example	58.3	55.6	63.1	60.8	63.2	61.2
PRAISE (50)	**60.3**	**57.3**	**65.6**	**63.2**	**66.0**	**63.9**

Table 4: Satisfaction classification performance: with vs. without passage generation

Impact of initial strategy selection on performance We investigated whether initial strategies influence subsequent performance. Although these



(a) MWOZ dataset (b) SGD dataset (c) ReDial dataset

Figure 6: Effect of exploration ratio on maximum test score across iterations. We experimented with exploration ratios of 0.0, 0.1, and 0.2, where an exploration ratio of 0.0 means not using the unorthodox planner at all.

strategies become less impactful throughout the training process, we considered it crucial to verify their potential long-term effects on the model's performance. We conducted an experiment using 5 sets of 5 randomly generated strategies for satisfaction classification, each trained 20 times over 50 steps. Our analysis, involving normality tests, homogeneity of variance tests, and ANOVA, revealed significant differences in average performance during the initial steps. However, these differences disappeared starting in the fourth iteration, with consistently high p-values in subsequent iterations. This finding suggests that while initial strategies influence performance in the early training stages, their impact diminishes as the PRAISE process continues. Further details on performance comparisons across different sets and steps can be found in Appendix C.

5 Related work

User satisfaction estimation Prior works on user satisfaction estimation in dialogue systems have evolved from content-based methods, such as sentiment analysis (Song et al., 2019) and interaction quality assessment (Schmitt and Ultes, 2015), to leveraging pre-trained language models (Kachuee et al., 2021) and dialogue action tasks (Deng et al., 2022), modeling satisfaction dynamics (Ye et al., 2023), and using multi-task adversarial method (Song et al., 2023). Recent efforts have explored using LLMs as simulators (Hu et al., 2023) and augmenting datasets with counterfactual dialogue samples (Abolghasemi et al., 2024). Despite recent advancements, many of these methodologies still struggle with interpretability. To address these limitations, Lin et al. (2024) proposed a framework using LLM, but it falls short in providing instance-level interpretability and incurs high costs when summarizing and classifying data using GPT-4. PRAISE achieves high efficiency by using LLM only to generate strategies during training, without requiring it for inference. Additionally, this approach provides interpretability for overall USE classification and utterance-level analysis.

Retrieval models Text embeddings that capture semantic similarity and context have been developed, ranging from early models (Liu et al., 2019; Mikolov et al., 2013; Pennington et al., 2014; Reimers and Gurevych, 2019) to larger models (BehnamGhader et al., 2024) trained on large-scale corpora. Diverse retrieval tasks utilize these embeddings, such as DPR (Karpukhin et al., 2020), Contriever (Izacard et al., 2021), ANCE (Xiong et al., 2020), and RocketQA (Qu et al., 2020). However, applying these retrievers to domain-specific data is challenging due to the lack of relevance supervision data. To address this issue, studies have utilized LLMs to generate pseudo-labels, such as hypothetical documents (Gao et al., 2022b) or generated relevant queries (Bonifacio et al., 2022; Boytsov et al., 2023; Dai et al., 2022; Jeronymo et al., 2023) for model training. To resolve the scarcity of relevance data, we employ LLMs to generate passages that align with the generated strategies. We then utilize the search results between these generated passages and actual user utterances as features in our framework.

6 Conclusion

In this paper, we presented PRAISE, a novel framework for User Satisfaction Estimation (USE) in dialogue systems that leverages Large Language Models (LLMs). Our work addresses the crucial challenge of developing interpretable and scalable methods for assessing user satisfaction, a key factor in improving conversational AI. PRAISE consists of three main modules: the Strategy Planner, which generates natural language strategies for classifying user satisfaction; the Feature Retriever, which provides multi-level interpretability by aligning user utterances with strategies; and the Score Analyzer, which evaluates strategy effectiveness and enables efficient inference by transforming LLM knowledge into a structured representation. Our experimental results across three benchmark datasets demonstrate PRAISE's superior performance compared to existing USE methods.

Limitations

PRAISE has several limitations that require further exploration, which can be summarized into three main points:

Embedding models. In PRAISE, we use a basic embedding model without fine-tuning. Enhancing the embedding model to better understand complex dialogue contexts and user intent, specifically for the USE task, could improve performance.

LLM-Driven Strategies. As the effectiveness of PRAISE Strategies is heavily dependent on the internal knowledge of LLM, their performance might be significantly compromised in domains where the LLM has limited or insufficient information, potentially leading to suboptimal or unreliable outputs. Future work should integrate external knowledge or additional modules to support strategy generation process.

Open-Domain Dialogues Evaluation. PRAISE currently focuses on task-oriented dialogue datasets, as these are the only datasets with user satisfaction annotations. In the future, evaluating PRAISE on open-domain dialogues, which more closely resemble real-world conversations, will be essential for expanding its practical applications.

Acknowledgements

This work was partly supported by the BK21 FOUR Program (Education and Research Center for Industrial Innovation Analytics) funded by the Ministry of Education, Korea

References

Amin Abolghasemi, Zhaochun Ren, Arian Askari, Mohammad Aliannejadi, Maarten de Rijke, and Suzan Verberne. 2024. Cause: Counterfactual assessment of user satisfaction estimation in task-oriented dialogue systems. *Preprint*, arXiv:2403.19056.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Praveen Kumar Bodigutla, Lazaros Polymenakos, and Spyros Matsoukas. 2019. Multi-domain conversation quality evaluation via user satisfaction estimation. *Preprint*, arXiv:1911.08567.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.

Leonid Boytsov, Preksha Patel, Vivek Sourabh, Riddhi Nisar, Sayani Kundu, Ramya Ramanathan, and Eric Nyberg. 2023. Inpars-light: Cost-effective unsupervised training of efficient rankers. *arXiv preprint arXiv:2301.02998*.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Wanling Cai and Li Chen. 2020. Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20, page 33–42, New York, NY, USA. Association for Computing Machinery.

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 225–234.

Jason Ingyu Choi, Ali Ahmadvand, and Eugene Agichtein. 2019. Offline and online satisfaction prediction in open-domain conversational systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1281–1290.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Preprint*, arXiv:1412.3555.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Yang Deng, Wenxuan Zhang, Wai Lam, Hong Cheng, and Helen Meng. 2022. User satisfaction estimation with sequential dialogue act modeling in goal-oriented conversational systems. In *Proceedings of the ACM Web Conference 2022*, pages 2998–3008.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022a. Precise zero-shot dense retrieval without relevance labels. *Preprint*, arXiv:2212.10496.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022b. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Zhiyuan Hu, Yue Feng, Anh Tuan Luu, Bryan Hooi, and Aldo Lipani. 2023. Unlocking the potential of user feedback: Leveraging large language model as user simulators to enhance dialogue system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 3953–3957, New York, NY, USA. Association for Computing Machinery.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R Lyu. 2019. Higru: Hierarchical gated recurrent units for utterance-level emotion recognition. *arXiv preprint arXiv:1904.04446*.

Mohammad Kachuee, Hao Yuan, Young-Bum Kim, and Sungjin Lee. 2021. Self-supervised contrastive learning for efficient user satisfaction prediction in conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4053–4064, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Runze Liang, Ryuichi Takanobu, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Minlie Huang. 2021. Turn-level user satisfaction estimation in E-commerce customer service. In *Proceedings of the 4th Workshop on e-Commerce and NLP*, pages 26–32, Online. Association for Computational Linguistics.

Ying-Chun Lin, Jennifer Neville, Jack W Stokes, Longqi Yang, Tara Safavi, Mengting Wan, Scott Counts, Siddharth Suri, Reid Andersen, Xiaofeng Xu, et al. 2024. Interpretable user satisfaction estimation for conversational systems with large language models. *arXiv preprint arXiv:2403.12388*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Preprint*, arXiv:1705.07874.

Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Yan Pan, Mingyang Ma, Bernhard Pflugfelder, and Georg Groh. 2022. User satisfaction modeling with domain adaptation in task-oriented dialogue systems. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 630–636, Edinburgh, UK. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Preprint*, arXiv:1602.04938.

Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: Assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication*, 74:12–36.

Darius Koenig Julius Lipp Sean Lee, Aamir Shakir. 2024. Open source strikes bread - new fluffy embeddings model.

Clemencia Siro, Mohammad Aliannejadi, and Maarten De Rijke. 2023. Understanding and predicting user satisfaction with conversational recommender systems. *ACM Transactions on Information Systems*, 42(2):1–37.

Kaisong Song, Lidong Bing, Wei Gao, Jun Lin, Lujun Zhao, Jiancheng Wang, Changlong Sun, Xiaozhong Liu, and Qiong Zhang. 2019. Using customer service dialogues for satisfaction analysis with context-assisted multiple instance learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 198–207, Hong Kong, China. Association for Computational Linguistics.

Kaisong Song, Yangyang Kang, Jiawei Liu, Xurui Li, Changlong Sun, and Xiaozhong Liu. 2023. A speaker turn-aware multi-task adversarial network for joint user satisfaction estimation and sentiment analysis. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Weiwei Sun, Shuo Zhang, Krisztian Balog, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. 2021. Simulating user satisfaction for the evaluation of task-oriented dialogue systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2499–2506.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Fanghua Ye, Zhiyuan Hu, and Emine Yilmaz. 2023. Modeling user satisfaction dynamics in dialogue via hawkes process. *arXiv preprint arXiv:2305.12594*.

Yue Yu, Siyao Peng, and Grace Hui Yang. 2019. Modeling long-range context for concurrent dialogue acts recognition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2277–2280.

A Prompt

A.1 Problem definition

Table 5 shows the common prompts and problem definition of each dataset.

A.2 Initial strategies

Table 6 shows the initial strategies for each dataset that we used in our experiment.

A.3 Strategy planner prompts

In all prompts, {{$variable}} acts as a placeholder to accept external variables.

For example, {{$problem_definition}} takes as input the problem definition for each dataset in Appendix A.1.

A.3.1 Great planner

```
hyperparameter : {
  "model": "gpt-4-1106-preview",
  "temperature": 0.1,
  "max_tokens": 512
}
```

Common prefix prompt
You are a competent bot that generates strategies to classify conversations in which the user expresses satisfaction.
MWOZ
The User and Assistant are having a conversation about making a reservation for a specific service, or looking up information such as an address or phone number. The types of services include taxis, restaurants, buses, hotels, attractions, and trains. The user asks a number of questions about the service, and their satisfaction depends on the assistant's answers. Users are satisfied if the assistant answers their questions appropriately, but they are also dissatisfied if the service provider does not provide the information they asked for, regardless of the assistant's answer.
SGD
Assistant is a virtual assistant that provides information about Alarm, Bank, transportation(bus, flight, etc.), reservation(rental car, restaurant etc.), Calendar, Event, Home, Hotel, Media, Movie, Music, Service, Travel, Weather and many other things people might want to know. A typical satisfaction for a user is when they successfully make a reservation or find the assistant's suggestions helpful, and sometimes they are dissatisfied with the assistant's answer and ask for another alternative or decline. Include specific context in your strategy for the information the assistant provides. (e.g. user requests a bus at a different time.)
ReDial
The user and the assistant have a conversation about movies, talking about the movies they've seen or recommending movies to each other. The Assistant's suggestions, questions, and reactions have a significant impact on the user's satisfaction, which can be inferred from the user's conversations. The main topics of conversation are the title, actors, and genre of the movie, but they also include casual conversation.

Table 5: Common prompts and problem definition of each dataset

MWOZ
• The user thanks the assistant. • The user repeats the same question. • The user asks about other services.
ReDial
• User asks for more movie recommendations. • User expresses interest in a movie's director. • User compliments assistant's choice. • User requests further details on movie. • User expresses interest in a specific genre.
SGD
• User expresses satisfaction with the service quality. • User acknowledges assistant's quick thinking. • User shows appreciation for assistance. • User empathizes with the assistant. • User appreciates the detailed explanation.

Table 6: Initial strategies of each dataset

system prompt

```
{{$problem_definition}}

[ouput format]
Your answer should be in the following json format
{
  "strategies": [
    "User [common verb] [appropriate object less than 5 words].",
    "User [common verb] [appropriate object less than 5 words].",
    ...
  ]
}
```

```
Below are the strategies created so far

[Effective strategies]
{{$effective_strategies}}

[Ineffective strategies]
{{$ineffective_strategies}}

Generate {{$strategy_num}} additional effective strategies that
you think would help your analysis.
answer:
```

A.3.2 Unorthodox planner

```
hyperparameter : {
  "model": "gpt-4-1106-preview",
  "temperature": 0.7,
  "max_tokens": 512
}
```

system prompt

```
[problem definition]
{{$problem_definition}}

[ouput format]
Your answer should be in the following json format
{
  "strategies": [
    "User [common verb] [appropriate object that fits the strategy].",
    "User [common verb] [appropriate object that fits the strategy].",
    ...
  ]
}

Below are the strategies created so far

[Effective strategies]
{{$effective_strategies}}
```

```
[Ineffective strategies]
{{$ineffective_strategies}}

In our opinion, the above strategies are too formulaic,
and sometimes crazy strategies that are completely weird
or nonsensical are more successful.

Generate {{$strategy_num}} strategies that sound like conversations
you'd have in a problem definition situation, but don't seem to have
anything to do with user satisfaction.

answer:
```

A.4 Feature retriever prompts

A.4.1 Passage generator

```
hyperparameter : {
  "model": "gpt-3.5-turbo-0125",
  "temperature": 0.0,
  "max_tokens": 1024
}
```

system prompt

```
[query]
{{$query}}

Create 5 messages that you think would come up as search results
if I were to search for messages that match the query.
The messages should be very natural, colloquial, and provided
in bullet type.
Answers should be of varying lengths, including short sentences of
two to three words and longer sentences using up to 10 words.

your answers:
```

A.5 GPT evaluation prompts

```
hyperparameter : {
  "model": "gpt-3.5-turbo-0125",
  "temperature": 0.0,
  "max_tokens": 128
}
```

system prompt

```
You are a competent bot that can look at a [conversation]
and determine whether the user at the
end of the conversation is satisfied or not.
Please answer "satisfied", "dissatisfied", or "neutral".
Don't answer anything else.

For each of the following criteria
  satisfied : The assistant's answer meets the user's needs
  and the user feels satisfied.
  dissatisfied : The user's needs are not yet met and they
  feel dissatisfied.
  neutral : neither of the above two cases, or simply
  informational or greeting.

[conversation]
{{$conversation}}

answer:
```

For few-shot, we added one sample each of SAT, NEU, and DSAT from the train dataset to the context.

B Interpretability

B.1 Additional examples demonstrating feature F interpretability

Table 7 illustrates how the feature values of individual user utterances can provide insights into the reasons behind the predicted satisfaction level, further highlighting the explanatory power of the PRAISE approach.

B.2 Additional box-plot analysis of relevance scores for strategies

In Figure 7, strategies that directly express satisfaction or gratitude, such as strategy_1 and strategy_5, exhibit a clear distinction in the SAT label. On the other hand, strategy_22, which involves asking for more details, shows higher scores in the NEU label compared to both SAT and DSAT.

Figure 8 demonstrates that, similar to other datasets, strategies with direct positive expressions yield high scores in the SAT label. Notably, strategy_17, which represents cases where users request further clarification, serves as a clear criterion for distinguishing the NEU label.

C Impact of initial strategy selection

Table 8 shows performance comparison across 5 sets of 5 randomly generated strategies for satisfaction classification, each trained 20 times over 50 steps.

step	set_1	set_2	set_3	set_4	set_5	p_value
1	0.5239	0.5349	0.5316	0.5244	0.5272	0.069874
2	0.5397	0.5508	0.5509	0.5452	0.5415	0.023499
3	0.5540	0.5618	0.5618	0.5609	0.5510	0.002639
4	0.5631	0.5654	0.5675	0.5661	0.5589	0.109610
5	0.5660	0.5667	0.5690	0.5697	0.5638	0.429035
...						
46	0.5900	0.5903	0.5941	0.5928	0.5920	0.480261
47	0.5914	0.5911	0.5939	0.5940	0.5928	0.679273
48	0.5911	0.5915	0.5955	0.5941	0.5940	0.378739
49	0.5916	0.5908	0.5952	0.5939	0.5930	0.447692
50	0.5917	0.5907	0.5953	0.5948	0.5928	0.370706

Table 8: Performance comparison across different sets and steps

D Different models for score analyzer

The core requirements for the Score Analyzer in the PRAISE framework are interpretability and a simple model capable of rapid training. We selected Random Forest and Logistic Regression as suitable models for these requirements and conducted experiments. The experimental results using Random Forest showed a 5$\sim$10% decrease in performance compared to our final PRAISE implementation. Based on these results, we ultimately selected Logistic Regression as it demonstrated superior performance while maintaining comparable interpretability to Random Forest.

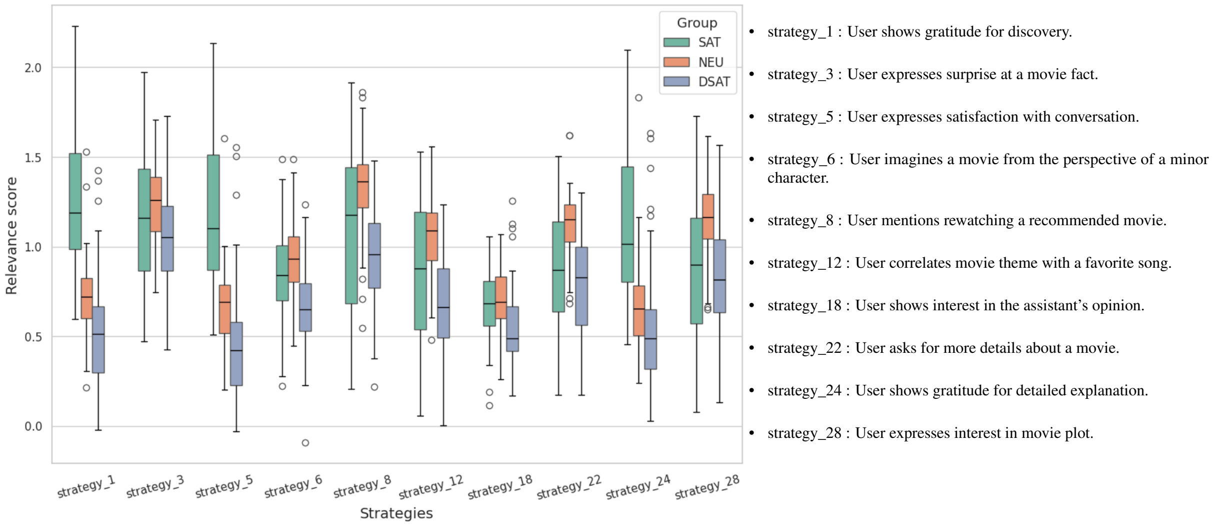


Figure 7: Box-plot of relevance scores for strategies in the ReDial dataset

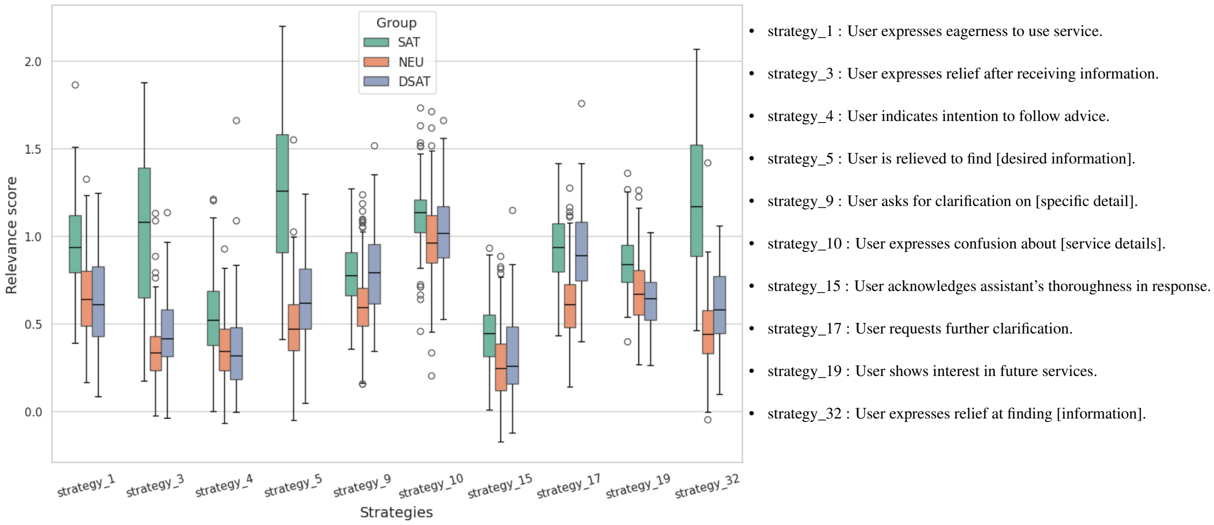


Figure 8: Box-plot of relevance scores for strategies in the MWOZ dataset

E License

The licenses for the datasets and baselines are as follows: ReDial (CC-BY-4.0), MWOZ (MIT License), SGD (CC-BY-SA-4.0), USDA (CC-BY-4.0), ASAP (MIT License).

Conversation **User**: I am trying to find a scary film to watch...I like a lot of styles but today I fancy a scary one **Assistant**: Have you seen "It (2017)"? **User**: That's a bit too scary for me. **Assistant**: How about "Dracula Untold (2014)" ? I hear it's pretty good **User**: oooh, i have not heard of "Dracula Untold (2014)" ...that sounds good...I like vampire movies.
Label : SAT
Reason (Score) User expresses joy over movie discovery. (1.82) User shows interest in movie plot. (1.77) User shows interest in the assistant's opinion. (1.02)

Conversation **User** : Do they offer daily housekeeping? **Assistant** : Daily housekeeping is not available at this hotel. Would you like me to book a room for you? **User** : Yes that will work, please book.
Label : SAT
Reason (Score) User appreciates assistant's recommendation (1.25) User expresses eagerness to use service (1.17) User thanks assistant for patience (1.14)

Conversation **User** : Thank you. Can you please confirm for me that the guesthouse you booked for me is moderately priced? **Assistant** : It is actually cheaply priced. Is that going to be okay? **User** : No, sorry. I am looking for something moderately priced.
Label : DSAT
Reason (Score) User expresses confusion about service details (1.04) User shows frustration over lack of information (0.84) User inquires about alternatives for service (0.82)

Conversation **User** : That suits me well. Can you tell me the address of the venue? **Assistant** : Your tickets have been bought. Enjoy your time at the event! The address of the venue is 24 Willie Mays Plaza. **User** : Can you tell me where will the event happen, and at what time it starts?
Label : NEU
Reason (Score) User requests additional information (1.30) User expresses excitement about the trip details (1.11) User confirms booking details eagerly (1.06)

Table 7: Examples of user utterances